```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, transit, finish int) {
    if n > 0 {
        Hanoy(n-1, start, finish, transit)
        fmt.Println(start, finish)
        Hanoy(n-1, transit, start, finish)
    }
}

func main() {
    Hanoy(8, 1, 2, 3)
}
```

01.classic

```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, transit, finish int) {
    if n > 0 {
        if start == 1 && finish == 3{
            Hanoy(n-1, 1, 2, 3)
            fmt.Println("1 2")
            Hanoy(n-1, 3, 2, 1)
            fmt.Println("2 3")
            Hanoy(n-1, 1, 2, 3)
        } else {
            Hanoy(n-1, start, finish, transit)
            fmt.Println(start, finish)
            Hanoy(n-1, transit, start, finish)
        }
    }
}

func main() {
    Hanoy(3, 1, 2, 3)
}
```

02.repair1

```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, transit, finish int) {
    if n > 0 {
        if (start == 1  && finish == 3) || (start == 3  && finish == 1) {
        // variant: if transit == 2
            Hanoy(n-1, start, transit, finish)
            fmt.Println(start, transit)
            Hanoy(n-1, finish, transit, start)
            fmt.Println(transit, finish)
            Hanoy(n-1, start, transit, finish)
        } else {
            Hanoy(n-1, start, finish, transit)
            fmt.Println(start, finish)
            Hanoy(n-1, transit, start, finish)
        }
    }
}

func main() {
    Hanoy(3, 1, 2, 3)
}
```

03.repair2

```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, transit, finish int) {
    if n > 0 {
        if (start == 1  && finish == 2)
        || (start == 2  && finish == 3)
        || (start == 3  && finish == 1) {
            Hanoy(n-1, start, finish, transit)
            fmt.Println(start, finish)
            Hanoy(n-1, transit, start, finish)
        } else {
            Hanoy(n-1, start, transit, finish)
            fmt.Println(start, transit)
            Hanoy(n-1, finish, transit, start)
            fmt.Println(transit, finish)
            Hanoy(n-1, start, transit, finish)
        }
    }
}

func main() {
    Hanoy(3, 1, 2, 3)
}
```

04.cyclic

```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, transit, finish int) {
    if n > 0 {
        Hanoy(n-1, start, finish, transit)
        fmt.Println(start, finish)
        Hanoy(n-1, transit, start, finish)
    }
}

func Hanoy2(n int, start int) {
    if n > 0 {
        switch {
        case start == 1 && n%2 == 0:
            Hanoy(n-1, 1, 2, 3)
            fmt.Println(1, 2)
            Hanoy2(n-1, 3)
        case start == 1 && n%2 == 1:
            Hanoy(n-1, 1, 3, 2)
            fmt.Println(1, 3)
            Hanoy2(n-1, 2)
        case start == 2  && n % 2 == 0:
            Hanoy2(n-1, 2)
        case start == 2  && n % 2 == 1:
            Hanoy(n-1, 2, 3, 1)
            fmt.Println(2, 3)
            Hanoy2(n-1, 1)
        case start == 3  && n % 2 == 0:
            Hanoy(n-1, 3, 2, 1)
            fmt.Println(3, 2)
            Hanoy2(n-1, 1)
        case start == 3  && n % 2 == 1:
            Hanoy2(n-1, 3)
        }

    }
}

func main() {
    Hanoy2(4, 1)
}
```

05.sorting

```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, finish int) {
    if n == 1 {
        fmt.Println(start, finish)
    }
    if n > 1 {
        Hanoy(n-1, start, finish)
        fmt.Println(start, 2)
        Hanoy(n-1, finish, start)
        fmt.Println(2, finish)
        Hanoy(n-1, start, finish)
    }
}

func main() {
    Hanoy(2, 1, 3)
}
```

06.unfair

```go
package main

import (
    "fmt"
)

func Hanoy(n int, start, transit, finish int) {
    if n == 1 {
        fmt.Println("  1:", start, finish)
    }
    if n > 1 {
        Hanoy(n-1, start, transit, finish)
        Hanoy(n-2, finish, start, transit)
        fmt.Println("swap", start, finish)
        Hanoy(n-2, transit, finish, start)
        Hanoy(n-1, start, transit, finish)
    }
}

func main() {
    Hanoy(4, 1, 2, 3)
}
```

07.exchanging